

# Workshop: Introductie PostgreSQL

---

## 1 Introductie

Als startpunt van dit vak zullen wij jullie in deze workshop wat algemene basiskennis aanrijken met betrekking tot de installatie van en de communicatie met databank-systemen. Het doel van deze workshop is namelijk om jullie vertrouwd te maken met de werking van een databankbeheersysteem (dbms in het kort).

Belangrijk om op te merken is dat we in deze workshop (en ook in de volgende workshops) regelmatig gebruik zullen maken van de commandolijn. Als je geen (of weinig) ervaring hebt met de commandolijn, kan je best, vooraleer je aan deze workshop begint, het document waarin de commandolijn geïntroduceerd wordt doorneemen. Dit document kan je terugvinden in de map van deze workshop op Ufora.

Concreet wordt er in deze workshop eerst dieper ingegaan op het databankbeheersysteem dat we doorheen dit semester vrijwel altijd zullen gebruiken, zijnde PostgreSQL (zie Sectie 2). We leggen uit wat PostgreSQL precies is en hoe je dit systeem kan installeren. Vervolgens leren we in Sectie 3 twee manieren aan om met een PostgreSQL databank te connecteren en te communiceren: via een commandolijnapplicatie (psql) en via een webapplicatie (pgAdmin 4). Tot slot tonen we je, in Sectie 4, hoe je een backup kan maken van een PostgreSQL databank en hoe je deze backup opnieuw kan inladen (restore).

## 2 PostgreSQL

In dit vak zullen we voornamelijk werken met het relationeel databankmodel, dat uitgebreid geïntroduceerd wordt in de theorielessen. Aangezien dit model enorm

populair is, bestaan er zeer veel verschillende commerciële implementaties van relationele databankbeheersystemen. Dergelijke programma's voorzien alle componenten van het relationele model (zoals tabellen, primaire sleutels, vreemde sleutels. . .) en laten dus toe om relationele databanken fysiek volledig te implementeren. Voor deze lessenreeks, hebben wij gekozen voor een van de meest gekende, en bovendien een open-source en gratis te downloaden relationeel databankbeheersysteem: PostgreSQL. Hoewel dit slechts 1 van de vele relationele databankbeheersystemen is, kan je veel van de aangeleerde concepten ook rechtstreeks toepassen op andere courante systemen, zoals MySQL, Oracle, SQL Server. . .

Om een relationele databank aan te maken, moeten we eerst en vooral de PostgreSQL dbms software downloaden en installeren. Hoe je dit doet, staat uitgebreid uitgelegd in een aparte installatiehandleiding die te vinden is in de map van deze workshop op Ufora. Volg nauwgezet de stappen die in deze handleiding beschreven staan om PostgreSQL, versie 16 (en het programma pgAdmin 4, waarmee je eenvoudig kan communiceren met PostgreSQL databanken) te installeren, vooraleer verder te gaan met deze workshop.

Installeer PostgreSQL, versie 16 en pgAdmin 4 op je eigen computer door de stappen in de installatiehandleiding te volgen.

### 3 Communiceren met een databank

Wanneer je de handleiding volledig hebt doorlopen, zou PostgreSQL op een correcte manier geïnstalleerd moeten zijn en zou er bovendien een lokale PostgreSQL cluster moeten draaien op jouw PC. Bij het opstarten van PostgreSQL opent er geen interactief scherm waarin je allerlei dingen kan doen, zoals je misschien gewoon bent na het opstarten van een programma. PostgreSQL is namelijk een programma (ook wel service genoemd) dat in de achtergrond draait op jouw PC. Tijdens de installatie van PostgreSQL wordt er gewoonlijk 1 databank en 1 gebruiker, beiden met naam postgres, aangemaakt op de cluster. Natuurlijk willen we ook andere databanken (en gebruikers) kunnen aanmaken en vervolgens kunnen gebruiken en beheren. Daartoe moeten we eerst verbinden met de PostgreSQL cluster waarin we deze databanken willen aanmaken. Om te kunnen verbinden met een PostgreSQL cluster dien je typisch een aantal connectieparameters op te geven die gekend zijn door deze PostgreSQL cluster. De belangrijkste connectieparameters zijn

- een IP-adres/hostname: het IP-adres of de hostnaam van de PC waarop de PostgreSQL cluster draait (jouw lokale PostgreSQL cluster draait op IP-adres 127.0.0.1, alias localhost),

- een poort: de poort van de host waarop je de PostgreSQL cluster kan bereiken (standaard ingesteld op 5432),
- een databanknaam: de naam van de databank waarmee je op de PostgreSQL cluster rechtstreeks wil verbinden (standaard ingesteld op postgres),
- een gebruikersnaam: de naam van de gebruiker waarmee je jezelf wil identificeren bij de PostgreSQL cluster (standaard ingesteld op postgres),
- een wachtwoord: het wachtwoord van de gebruiker waarmee je jezelf wil identificeren bij de PostgreSQL cluster.

Merk dus op dat je heel eenvoudig kan verbinden met de cluster die lokaal op jouw eigen PC draait (via de standaard parameterwaarden). Onthoud echter dat je via jouw PC ook relatief eenvoudig kan verbinden met PostgreSQL clusters die op andere PCs (of servers) draaien.

Van zodra je bent verbonden met een cluster, kan je hier eenvoudig mee communiceren door instructies op te stellen die het dbms moet uitvoeren. Een van de grote voordelen van het relationele model is dat er een gestandaardiseerde taal bestaat om met relationele databankbeheersystemen te communiceren. Deze taal draagt de naam *Structured Query Language*, of kortweg SQL. De diverse commerciële implementaties gebruiken elk een eigen SQL-dialect, maar de communicatieconcepten waarmee je tijdens de komende lessen kennis zal maken en specifiek zijn voor PostgreSQL zijn dus ook grotendeels bruikbaar in het geval van andere implementaties. Er bestaan heel wat verschillende manieren om SQL-instructies door te geven aan PostgreSQL. In deze sectie zullen we twee manieren aan jullie voorstellen, zijnde psql (een commandolijn-applicatie) en pgAdmin, versie 4 (een webapplicatie met grafische interface). We zullen tonen hoe je, via deze twee communicatietools, met een PostgreSQL cluster kan verbinden, en hoe je een databank kan aanmaken en nadien opnieuw kan verwijderen.

### 3.1 Commandolijn

Een eerste manier om met een PostgreSQL databank te communiceren is door te werken met de psql-commandolijnapplicatie, die samen met PostgreSQL wordt geïnstalleerd. Zoals je ondertussen wel al weet, kan je in de commandolijn allerlei programma's uitvoeren die op jouw PC geïnstalleerd staan, waaronder dus ook psql. Voer, om de applicatie te starten, het volgende commando uit in de commandolijn<sup>1</sup>.

---

<sup>1</sup>Indien je een melding krijgt dat dit programma niet gevonden wordt, controleer dan of je PATH-variabele correct is ingesteld (zie installatiehandleiding). Als dit niet blijkt te werken, kan je nog steeds via de commandolijn naar de map navigeren waarin psql is geïnstalleerd en psql daar uitvoeren. Op Windows is dit typisch de map C:\Program Files\PostgreSQL\16\bin. Typ, daarnaast, het gegeven commando over en kopieer dit niet rechtstreeks vanuit dit document.

```
psql --host=127.0.0.1 --port=5432 --username=postgres --dbname=postgres
```

Door het bovenstaande commando uit te voeren, verbind je dus, via de psql-applicatie, met jouw lokaal draaiende PostgreSQL cluster. Merk op dat dit commando de hierboven vermelde connectieparameters als opties verwacht, met uitzondering van het wachtwoord. Naar dit wachtwoord wordt pas expliciet gevraagd na uitvoering van dit commando<sup>2</sup>.

Stel dat je, als `sql_exerciser` gebruiker, wil verbinden met de `exercises` databank die geïmplementeerd is op de PostgreSQL cluster die draait op de server met naam `ddcmstud.ugent.be` en bereikbaar is via poort 8088. Geef het volledige psql-commando dat je kan gebruiken om deze verbinding tot stand te brengen. Je hoeft dit commando niet uit te voeren.

Eens je succesvol geconnecteerd bent, kan je door uitvoering van het psql-commando `\l` (waarbij de 'l' voor 'list' staat) een overzicht opvragen van alle databanken die bestaan op jouw lokale PostgreSQL cluster. We willen nu echter een nieuwe databank, met naam `wielrennen`, aanmaken. Deze databank zullen we uitvoerig als leidraad tijdens de volgende oefeningenlessen zullen gebruiken. Het aanmaken van een databank kan als *SQL-instructie* doorgegeven worden aan het databankbeheersysteem. De PostgreSQL-instructie om een databank met naam `wielrennen` aan te maken is de volgende<sup>3</sup>.

```
CREATE DATABASE wielrennen;
```

Voer bovenstaande SQL-instructie uit en controleer opnieuw welke databanken er bestaan op je lokale PostgreSQL cluster.

Als alles goed gegaan is, zou de `wielrennen` databank nu in deze lijst moeten voorkomen. Nu de databank is aangemaakt, zouden we kunnen beginnen met het implementeren van de verschillende componenten in deze databank. Hiervoor moet je eerst connectie maken met de `wielrennen` databank die je juist hebt aangemaakt. Dit kan eenvoudigweg door middel van volgend psql-commando.

```
\c wielrennen
```

---

<sup>2</sup>Wanneer je in de commandolijn een wachtwoord intypt, zal je geen output zien verschijnen, ook al ben je wel degelijk de tekens die je intypt aan het doorgeven.

<sup>3</sup>Sluit, naar goede gewoonte, steeds alle SQL-instructies af met een puntkomma.

waarbij 'c' staat voor 'connect'. We tonen echter eerst hoe je een databank kan verwijderen, en zullen vervolgens met pgAdmin 4 de databank opnieuw aanmaken. Een databank kan met de volgende SQL-instructie worden verwijderd.

```
DROP DATABASE wielrennen;
```

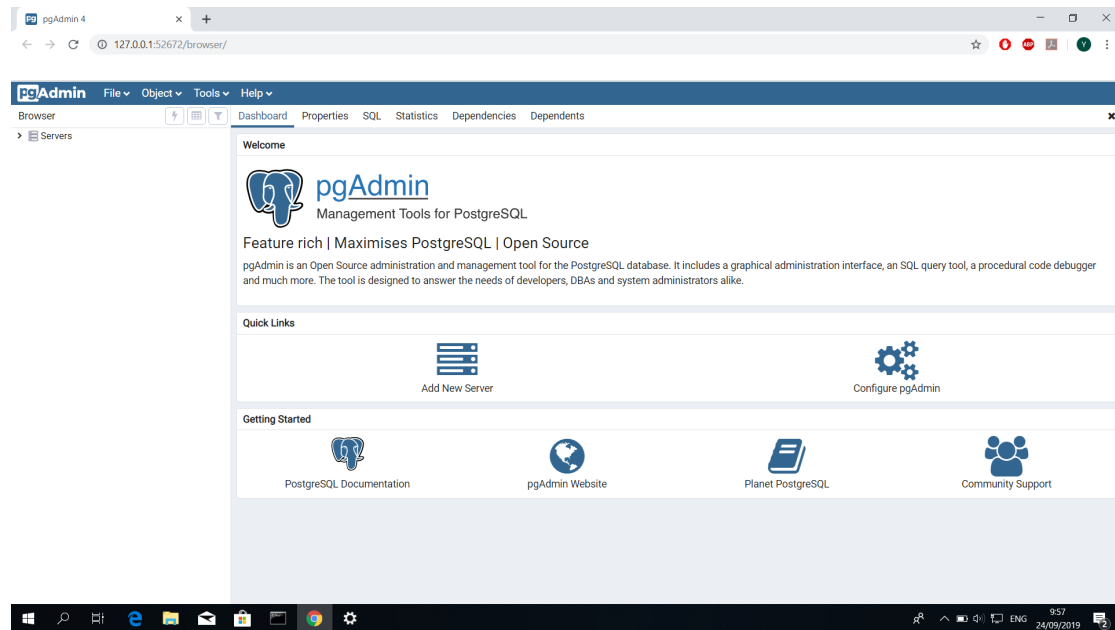
Voer bovenstaande SQL-instructie uit vooraleer verder te gaan met deze workshop. Let wel op, je kan geen databank verwijderen indien er nog connecties bestaan naar deze databank. Connecteer dus eerst terug met de postgres databank en verwijder alle openstaande connecties naar de wielrennen databank.

Voor de volledige documentatie van psql kan je steeds terecht op <https://www.postgresql.org/docs/current/app-psql.html>.

### 3.2 pgAdmin 4

Zoals we eerder al vermeld hebben is ook pgAdmin 4 een tool die gebruikt kan worden om te connecteren en te communiceren met PostgreSQL databanken. Net zoals in het geval van psql kan je door middel van pgAdmin 4 verbinden en communiceren met PostgreSQL databanken die op jouw lokale machine bestaan alsook met PostgreSQL databanken die op een andere machine geïmplementeerd zijn. Bovendien heeft de tool, in tegenstelling tot psql, een grafische interface waardoor je heel intuïtief met PostgreSQL databanken kan werken. Daarnaast biedt pgAdmin 4, net als psql, ook een interface aan om rechtstreeks SQL-instructies (bv. CREATE DATABASE ...) naar een PostgreSQL cluster te sturen. In pgAdmin 4 bestaan er met andere woorden twee manieren om met een PostgreSQL databank te interageren. We tonen beide manieren hieronder.

Na het opstarten van pgAdmin 4 opent er een venster waarin de applicatie je begroet met een beginscherm dat eruit ziet zoals het scherm dat wordt weergegeven in Figuur 1. Dit scherm is opgedeeld in 2 delen: een browservenster links en een detailgedeelte rechts. Het detailgedeelte toont steeds informatie over het deel van de PostgreSQL cluster dat links geselecteerd is. Ook kan je in dit detailgedeelte acties uitvoeren die inwerken op het geselecteerde deel van de PostgreSQL cluster. Net zoals in het geval van psql zullen we eerst een connectie moeten maken met de PostgreSQL cluster die lokaal draait op jouw PC (hostname localhost, poort 5432), vooraleer we een nieuwe databank met naam wielrennen kunnen aanmaken. In het linkervenster wordt er een overzicht weergegeven van alle bestaande connecties. Een connectie maken met jouw lokale PostgreSQL cluster kan je doen op de manier die beschreven staat in onderstaande opdracht.



Figuur 1: pgAdmin 4 beginscherm.

Maak een connectie met de 'localhost' cluster door in het browservenster met de rechtermuisknop te klikken op Servers - Register - Server... Geef de aangemaakte connectie de naam 'local'. De waarden voor de verschillende connectieparameters (hostname, poort, databanknaam...) kan je invoeren onder het 'Connection' tabblad.

Nu kan je de 'local' verbinding openklikken. Onder 'Databases' vind je dan alle databanken die op deze cluster bestaan. Aangezien we daarnet via psql de aangemaakte wielrennen databank opnieuw verwijderd hebben, zie je deze databank niet meer in het overzicht staan.

Maak de wielrennen databank aan via de grafische interface van pgAdmin 4 door rechts te klikken op Databases - Create - Database... en vervolgens de naam van de databank in te vullen.

Normaal gezien staat de databank nu zichtbaar tussen de andere databanken die op

jouw lokale PostgreSQL cluster bestaan<sup>4</sup>. We zullen echter de databank ook eens aanmaken door in pgAdmin 4 rechtstreeks SQL-instructies door te geven.

Verwijder de wielrennen databank door rechts op de naam van deze databank te klikken en 'Delete' te selecteren.

In pgAdmin 4 kan je dus ook rechtstreeks SQL-instructies doorgeven aan het dbms, net zoals we dit eerder deden in psql. Dit kan via de 'query tool' die je opent door rechts te klikken op de naam van een (willekeurige) databank in het browservenster, en dan 'Query Tool' te selecteren.

Maak de wielrennen databank aan door middel van de CREATE DATABASE SQL-instructie en klik vervolgens op het uitvoeringssymbool bovenaan in het venster of druk op F5 om de instructie uit te voeren.

We hebben nu op drie verschillende manieren de wielrennen databank aangemaakt. In het vervolg zullen we gebruik maken van pgAdmin 4 vanwege de grafische ondersteuning, maar we zullen wel met SQL-instructies blijven werken. Het is immers van cruciaal belang om SQL goed onder de knie te krijgen, zodat je niet afhankelijk bent van de grafische interface van pgAdmin 4 om met een (PostgreSQL) databank-beheersysteem te communiceren. Dit zorgt er dan ook voor dat je later met andere tools een (PostgreSQL) databank kan beheren.

## 4 Backup en restore

Databanken en hun data worden opgeslagen in het permanente geheugen (de harde schijf) van de machine waarop deze databanken zijn geïmplementeerd. Dit zorgt ervoor dat de betrouwbaarheid van de opslag rechtstreeks afhankelijk is van de hardware. Als je niet de juiste voorzorgsmaatregelen neemt, bestaat de kans dat op een dag de harde schijf van deze machine crasht en je alle gegevens kwijt bent. Om dergelijke scenario's te voorkomen en om de inwisselbaarheid van data(banken) over verschillende machines te vereenvoudigen, voorziet PostgreSQL een backup-mechanisme. Dit mechanisme vertaalt een databank naar een reeks SQL-instructies

---

<sup>4</sup>Mogelijks moet je de pagina eens verversen vooraleer nieuw aangemaakte componenten worden weergegeven in het browservenster. Dit kan je doen door met de muis rechts op een component in het browservenster te klikken en 'Refresh...' te selecteren.

die nodig en voldoende zijn om deze databank later volledig te kunnen herimplementeren. Dit is niet alleen handig uit veiligheidsoverwegingen, maar ook om een databank van één machine naar een andere machine te kopiëren, zonder dat je daarvoor elke instructie op de nieuwe machine expliciet moet uitvoeren en je vervolgens alle data in de nieuwe databank opnieuw moet inladen. In het vervolg zullen we tonen hoe je een backup maakt van de wielrennen databank en hoe je deze backup ook weer kunt inladen via de commandolijnapplicaties `pg_dump` en `psql`.

## 4.1 Backup

Met de commandolijn-applicatie<sup>5</sup> `pg_dump` kan je een backup (oftewel een dump) maken van een bestaande databank. Deze applicatie kan je, net als `psql`, enkel rechtstreeks op de commandolijn uitvoeren (en dus niet binnen de `psql`- of `pgAdmin`-omgeving). `pg_dump` zorgt ervoor dat alle instructies die nodig zijn om de PostgreSQL databank waarvan je een backup maakt weggeschreven worden naar een `.sql`-bestand. Dit (backup) bestand bevat dus niets anders dan een sequentie SQL-instructies die je zou moeten uitvoeren om, vertrekkende van een lege toestand, te eindigen met een databank zoals die eruit ziet op het moment van de backup. Hieronder wordt een voorbeeld gegeven van hoe je het `pg_dump` commando kan oproepen.

```
pg_dump
--host=127.0.0.1
--port=5432
--dbname=wielrennen
--username=postgres
--create
--clean
--if-exists
--schema-only
--file=wielrennen.sql
```

Zoals je kan vaststellen, kan je heel wat verschillende opties meegeven aan het `pg_dump`-commando, waarvan je er reeds een aantal hebt leren kennen (zie Sectie 3.1). Aan de optie `--dbname` geef je de naam van de databank mee waarvan je een backup wil maken. Deze databank dient te draaien op de server met het IP-adres (of hostname) meegegeven aan de `--host` optie en op de poort meegegeven aan de `--port` optie. De opties `--clean` en `--if-exists` zorgen er voor dat de `DROP ...IF EXISTS` instructies worden toegevoegd aan het `.sql`-backupbestand om eerst alle bestaande componenten te verwijderen vooraleer de componenten terug aan te maken bij het inladen van de backup. Deze opties zijn handig bij het overschrijven

---

<sup>5</sup><https://www.postgresql.org/docs/current/app-pgdump.html>



van een bestaande databank. De optie `--create` zorgt ervoor dat in het backup-bestand ook de `CREATE DATABASE` instructie wordt opgenomen. Met deze instructie wordt een nieuwe, lege databank met de gespecificeerde naam aangemaakt. Met de `--schema-only` optie geef je aan dat je enkel de definitie van de databank wil dumpen, en niet de opgeslagen data. Met de optie `--username` geef je aan via welke gebruiker je een backup wil maken. Tenslotte geeft je na de `--file` optie de pad-naam op van het backup-bestand. Let op dat je het volledige commando op 1 lijn typt, met spaties om de verschillende opties te scheiden (dus niet met ‘newlines’ of ‘enters’ tussen de verschillende opties, zoals wordt weergegeven in bovenstaand voorbeeld).

Maak een *schema*-backup van de wielrennen databank waarin de databank *niet* opnieuw wordt aangemaakt. Je moet dus zelf nadenken welke opties er meegegeven moeten worden aan het `pg_dump` commando. Controleer de inhoud van het resulterende backup-bestand om te zien of dit gelukt is.

## 4.2 Restore

Via de commandolijn-applicatie `psql` is het mogelijk om een backup van een databank terug in te laden (restore). Dit kan je doen aan de hand van het volgende commando.

```
psql
--host=127.0.0.1
--port=5432
--dbname=wielrennen_backup
--username=postgres
--file=wielrennen.sql
```

Dit commando is gelijkaardig aan de `pg_dump` en `psql` commando's die reeds uitgebreid zijn toegelicht hierboven. We gaan dus geen verdere uitleg over de verschillende opties verschaffen. Let op, de `--dbname` optie kan je enkel gebruiken indien je backup-script geen `CREATE DATABASE` instructie bevat en je reeds een (lege) databank met de opgegeven naam hebt aangemaakt op de gegeven PostgreSQL server.

Maak via `psql` een tweede databank aan met naam `wielrennen_backup`. Verlaat vervolgens het programma `psql` (via `Ctrl + C` of `\q`) en restore de databank naar `wielrennen_backup` waarvan je een backup hebt gemaakt in Sectie 4.1

